UDC 044.9. 629.7.05 DOI: 10.18372/2073-4751.80.19768

Kovalenko Y.B., Candidate of Pedagogic Sciences,
orcid.org/0000-0002-6714-4258,
e-mail: yleejulee22@gmail.com,
Kozlyuk I.O., Doctor of Technical Sciences,
orcid.org/0000-0001-8239-8937,
e-mail: avia_ira@ukr.net

METHOD OF DEVELOPING AUTOMATED INFORMATION SUPPORT FOR INTEGRATED MODULAR AVIONICS SYSTEMS: MONITORING, DIAGNOSTICS AND RELIABILITY IMPROVEMENT

State University "Kyiv Aviation Institute"

Introduction

The avionics software development process consists of four detailed processes:

- a software requirements process that leads to the development of high-level requirements (HLR);
- a software design process that develops low-level requirements (LLR) and software architecture based on the HLR;
- the coding process that leads to the creation of source code and non-integrated object code;
- the process of software integration, which involves the consolidation of software in the form of executable programs and its integration with external devices.

High level requirements (HLR) are implemented based on system architecture and system requirements. These include timing signals, memory management, scheduled communications with external devices, error response and detection methods, system performance monitoring, and software separation.

The HLR is the basis for developing low-level requirements used in the software design process, which include descriptions of connections to external devices, definitions and how data flows, communication mechanisms, and software components.

The encoding process involves translating the LLR into source code and precompiled object code. This is related to the verification process, since at this stage the partially developed code is pre-executed. The integration process involves compiling and combining the compiled code into executable programs and embedding this software on the target platform.

Literature analysis and problem statement

Integrated modular avionics (IMA) is a key architecture in modern aircraft design, which allows reducing the number of autonomous subsystems by consolidating their functions on shared computing resources. The main advantage of IMA is the ability to share hardware resources, which increases reliability, efficiency of systems and reduces maintenance costs [1-5]. According to Collier et al. (2005), IMA revolutionized civil aviation by enabling multiple avionics applications to run concurrently on a single platform, increasing both productivity and flexibility [7]. Further research by Kinnan (2013) emphasizes that modularity within the IMA simplifies system upgrades and maintenance, while Heckmann (2009) describes how the IMA architecture facilitates systems engineering and integration in avionics systems by providing a robust framework for managing multiple systems [11-14]. Automation has become central to software development for IMA, particularly through the use of model-based modeling (MBD). MBD allows the automatic generation of software components, which reduces development time and ensures consistency between different systems. Tools such as Simulink and Embedded Coder play an important role in the development of real-time systems for avionics. Perry et al. (2014) investigated how MBD optimizes design and verification processes, while Grzelak et al. (2018) demonstrated that using Simulink for automatic code generation significantly improves software reliability and test efficiency, especially in complex and safety-critical avionics systems. The effectiveness of IMA systems is closely related to the reliability of the information support they provide during operation. Real-time data processing, system redundancy, and fault tolerance are important characteristics of modern avionics systems [15, 16].

Shames (2010) emphasized that effective information management practices are necessary for the seamless integration of various hardware and software components, which is critical for real-world decision-making. In addition, Chen et al. (2015) pointed out the need for fault-tolerance mechanisms that improve the reliability of IMA systems, especially under high-load conditions during operation. One of the main challenges in software development for IMA is to ensure compliance with security standards such as DO-178C and ARINC 653 It is important that automated systems meet these standards, especially given the increasing complexity of modern aviation systems. Hecht and Adelstein (2013) discussed the problems of certification of IMA systems, and Blanquart et al. (2017) emphasized the importance of automated safety assessment systems that can facilitate the verification and confirmation of safety compliance in complex avionics environments [17, 18].

Number of key challenges accompanies setting the problem of software development for automated complex and modular information equipment of avionics. First, the complexity of modularity requires seamless integration between many components, especially in real-world operating conditions. Second, implementing automated software development using model-based modeling raises concerns about ensuring compliance with strict security certification standards such as DO-178C. Third, the growing need for real-time data management creates additional requirements for existing computing platforms, especially for providing fault tolerance and system redundancy. Software development for automated complex and modular avionics information equipment faces significant challenges related to the integration of modular components, compliance with security certification standards, real-time data management and cyber security. It is necessary to develop improved methods, techniques and tools that will ensure seamless integration, high compliance with security standards and consideration of potential cyber threats. This problem statement lays the groundwork for further research into innovative methods of automation, security verification, and real-time systems development for aviation software.

The purpose and objectives of the study

Analysis of software development processes involves the study and assessment of system requirements at various levels. System requirements are divided into high-level and low-level requirements. High-level requirements are developed based on the system architecture and define the main characteristics of the software. In turn, low-level requirements detail these characteristics, turning them into specific technical solutions [10].

The task of analysis is to ensure that the source code meets both high- and low-level requirements. This is achieved by using the results of the software development life cycle to create clear and structured software requirements.

The main stages of the analysis include:

1. Statement of high-level requirements: includes a description of the distribution of system requirements, consideration of security requirements, functional and operational characteristics for each mode of operation, as well as performance criteria, requirements for execution time, memory, interfaces and other constraints.

2. Software Architecture Development: Based on the high-level requirements, a software architecture is created that meets the low-level requirements. This stage involves the creation of algorithms, data structures, input/output description, and resource management.

3. Assessment of compliance with requirements: At this stage, a check is carried out whether the software meets the

established requirements, as well as detection of possible errors, determination of methods for their detection and security monitoring.

Thus, the analysis allows you to check the correctness of software development, ensuring its compliance with requirements at all levels and contributing to the improvement of the quality, reliability and security of the software product.

The description should include intraprocessor and intratask communication mechanisms, including fixed interrupt timings, design methods, and implementation details [6]. An important element of the description is user-modified software, partitioning methods and measures to prevent partitioning, as well as a description of the software components and a reference to the base version from which they were created downloaded. This description should also include derived requirements arising from the software development process. If the system contains inactive code, the description of the security measures to activate the code on the target computer and the justification of the design decisions are directly included in the system requirements related to its security.

The software development process is complete when its goals and the goals of the associated integration processes are achieved. In the software coding process, the source code is implemented based on the software architecture and low-level requirements. The inputs to the coding process are the low-level requirements and software architecture from the software design processes, software development plan, and software coding standards. The software coding process can be started when the planned transition criteria are met. The source code is developed during this process and is based on the system architecture and low-level requirements. The target computer and the source code of the software coding processes are used to compile, merge and load the data in the integration process; it is aimed at the integration of the avionics system or its component equipment [9].

Developed algorithms for controlling the computer system of integrated modular avionics in the process of design and operation.

The avionics software integration process consists of four detailed processes:

- communication in the framework of software certification;
- management of requirements within the framework of software certification;
- verification within the framework of software certification;
- quality assessment as part of software certification.

Certification communication is an important process for the successful completion of software certification. This involves constant cooperation and communication between the applicant and the certification body. An applicant is an organization seeking certification. This process spans the entire software life cycle, starting with planning and ending with disposal. The applicant's task is to define conformity measures that define the manner in which the software will satisfy the essential requirements of the certification.

The requirements management process, like certification communication, spans the entire software lifecycle. It covers all data and documentation used for software development and testing. Requirements management is the "art" of defining, organizing, and controlling change during software development. The main task of this process is to achieve the maximum possible efficiency while minimizing errors. The requirements management method is related to the level of damage to the on-board equipment. The DO-178C standard defines two levels of software control: levels CC1 and CC2. The CC1 level must meet all the requirements of DO-178C, while the CC2 level only meets some of them.

The software verification process involves identifying and describing errors that have occurred from the software planning stage to the development stage. The DO-178C standard does not specify the methods to be used for verification, but rather the objectives to be achieved.

The task of the quality assessment process is to demonstrate that the developed software meets the expected requirements and standards, which, as a result, should lead to the fact that the product meets the expectations of the customer. Software quality assessment is a continuous process that begins in the planning phase and continues through the development and testing phases to the final product.

All of the above characteristics can be satisfied with respect to the developed avionics software by using appropriate automated control.

Proposed methodology

A method for predicting avionics software vulnerabilities using branching processes. One of the latest approaches to combating avionics software vulnerabilities is the use of vulnerability prediction models (VPMs). These models are based on machine learning elements, which allows predicting software components that may contain vulnerabilities in their future versions. VPM modules use software attributes from their previous versions as input, which are then used in a binary classification [8].

The most common learning techniques used to model software vulnerabilities are, for example, logistic regression, decision trees, k-nearest neighbors, naive Bayes, random forest, and support vector machine (SVM).

The conducted analysis indicated the two most popular types of VPM:

Using software metrics that take into account a certain set of software metrics when creating a binary classifier. The purpose of pretesting is to empirically assess and confirm expert opinion that software complexity is contrasted with software security. However, the overall weak relationship between complexity and security vulnerabilities leads to the need to explore different models to predict vulnerabilities, such as code modification, communication, code coverage, conjugation, sequence, and developer activity.

Use of text research methods, when the source code of tested software components is analyzed and presented in the form of a set of tokens. Tokens are combined into a dataset and a user along with vulnerability data to train vulnerability predictors. During the second phase, the trained classifier uses these datasets to determine whether the future version of the code module under study is vulnerable to bugs and hacking attacks or not.

Finding vulnerabilities in avionics security software that lead to bugs or hacking attacks is considered a worthwhile activity that can greatly impact flight safety and reliability. The ability to predict the occurrence of software vulnerabilities or quantify their impact enables you to predict software security trends and plan a well-understood security management process.

The developed method is aimed at improving the ability to predict vulnerabilities in tested software. Validation and further improvement of the accuracy of the proposed method requires further research with further empirical analysis using data from vulnerability databases or other types of vulnerability resources.

Therefore, one of the solutions implemented at AFIT in the area of error limitation in the developed avionics software is an automated control system in accordance with the requirements of the DO-178C standard and the implementation of these requirements in the form of a procedure in the ISO-9001 Quality Assurance System. The created computer system allows you to perform inspections and create documents required by the DO-178C standard (ie plans, reports, reports) directly from the IT network.

The task accepted for implementation was to create an automated software development management tool for flight parameters to be developed and built at AFIT according to the DO-178C standard.

The main functions of the built IT computer support system included:

- setting up a new project, which involves entering information about the name of the project, personal data of the project manager and individual contractors, their powers and access levels of the system to the system;
- entering data into the knowledge base regarding the implementation of the project (details, finances, restrictions);
- automatic generation of document templates required in the DO-178C standard

(i.e. plans, standards, procedures and inspection methods, reports and other records);

- automatic generation of tests for developed software and archiving of test results;
- archiving of correspondence between project executors, program files and the results of their testing;
- automatic backup of files to a server located in another building on the territory of AFIT (protection against data loss);
- provision of data on the implementation of the project according to the entered authorization of system users;
- project status reporting for audit or verification purposes in accordance with entered instructions.

A computer system supporting the management of the avionics software, after the installation of specialized software including static and dynamic analyzes adapted to the vulnerability of the software and error detection, to determine its vulnerability to structural damage of the avionics system and hacking attacks, provides direct, electronic cooperation between project participants and its control by the project manager, who is responsible for the correct implementation of the project.

Structural diagram of the computer system for supporting the management process

The main structural element of the computer system that provides management of avionics software development is a special server built into the AFIT IT network. The server interacts with the workstations of individual users of the system. A file server, called a backup, is used to protect the collected information. Its task is to archive files. The status of the project is stored in its memory after each "working day". It also cooperates with a failover server, which is activated in case of failure of the main server (protecting the current progress of the project).

The operating software of the main server includes the Windows Server operating system, the Windows SQL Server database, and the MS Office editing and calculation package as specialized software for project management using instructions according to the DO-178C standard.

A model of a computer system supporting the development of safety-critical avionics software; (1) a diagram of the architecture of a computer system integrated with an IT network; (2) file server architecture diagram.

Specialized software installed on the server uses computing modules, which include:

- basic analysis for preliminary testing and verification of the developed software;
- advanced analysis for preliminary testing and verification of developed software (Modified Condition/Decision Coverage, Information Flow Analysis, Dynamic Data Flow Coverage, Extract Semantic Analysis);
- additional analysis for preliminary testing and verification of the developed software;
- additional software for testing, which provides constant support for the development process of individual software components.
- specialized software integrated with a computer system to support the avionics software development process is presented.

In the technical implementation of the computer system to support the avionics software development management process, specialized modular computers were used, which functioned as servers and were embedded in the IT cabinet.

The AFIT IT network also includes power modules, switches, patch panels, cabling and accessories. The modular structure of servers and the use of connection panels allows you to choose the configuration of the IT network that is optimal for the system administrator and users.

Flight safety requirements are the result of a safety level assessment that includes functional, integration and reliability requirements for a given system. Error rate requirements are defined during the security assessment process to ensure system integrity by defining system defenses and responses to such errors. These requirements are defined for software and hardware to eliminate or limit the effects of errors, and to provide error detection, tolerance, removal, and avoidance. The system processes responsible for improving and assigning system requirements to hardware and/or software lead to the development of an appropriate architecture for the flight parameter display system.

BIOS configuration is understood as the process of adjusting the BIOS (Binary Input/Output System) parameters available to the programmer, resulting in improved interaction between the hardware and software layers [19].

Software whose non-standard behavior, as determined by the safety assessment process, could result in or contribute to a significant error, resulting in conditions that limit the functionality of the helicopter or add additional burden to the pilot, is subject to special analysis.

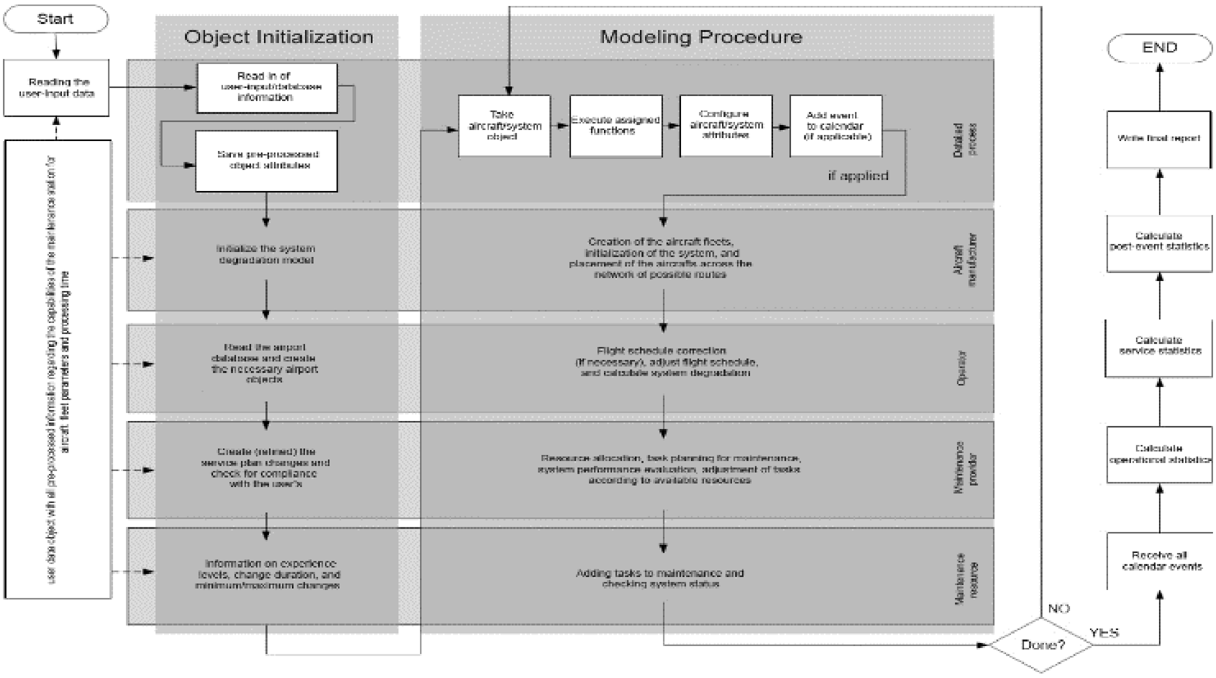

Fig. 1. The algorithms for controlling the computer system of integrated modular avionics in the process of design and operation

The planning process involving the software for the installation of the flight parameter display system is defined in such a way that the requirements are met and the level of reliability is adequate to the level of reliability of the adopted software.

According to the requirements of the DO-178C standard, software development processes, display of flight parameters are contained in the software planning process and the software development process. The processes involved in software development include the processes of defining software requirements; processes related to software coding and processes related to integration.

The software integration process for the flight parameter display system includes software integration and hardware/software integration. Integration processes can be performed when the planned transition requirements are met. The inputs of the integration process are the software architecture from the software design processes and the source code from the software coding processes, while the outputs of the integration process are the compiled object code files. Integrative processes are complete when their goals and the goals of related integrated processes are achieved. Object code must be generated from source code and then compiled.

All data parameter files must be generated and the software must be integrated into the host computer, target device emulator, or target device. The software must be implemented on the target computer for the purpose of hardware/software integration. Inconsistent or erroneous inputs identified during the integration process should be directed to the software requirements processes, software development processes, coding processes, or software planning processes as feedback that requires validation.

The graphics computer of the flight parameters display system transmits information about the flight parameters to the day display or the night display. The information received from the signal matching system, the GPS satellite navigation receiver and the aerodynamic data unit ADU is presented in the form of graphic symbols or in digital form.

The structure of graphic software consists of the following elements:

- direct setting of the BIOS processor operating system;
- configuration of the built-in WINDOWS XP operating system;
- configuration of the graphics computer user software;
- architecture of user software of a graphics computer.

A computer's graphics software is built into the non-volatile memory of the CPU motherboard. Calibration results for individual measurement channels are stored in the computer's external memory in a FLASH package.

Graphical identification of computer software includes information such as the name of the software, the software identification number, the software version identifier, the software component modules, and the license granted.

The system software is tested to demonstrate that it meets the essential requirements set forth in AQAP 2210 and DO-178C and to demonstrate with a high degree of confidence that errors that could lead to unacceptable failure conditions as defined in the ARP 4761 Security Assessment Process have been removed.

Numerical expression

The goal of computer software graphical testing is to demonstrate that object executable code, program code implemented directly in the device, satisfies high-level requirements and is closely related to low-level requirements.

Three types of tests have been identified for computer graphics software:

- hardware/software integration testing in order to verify the correct operation of the software on the target computer;
- software integration testing in order to verify the relationships between software and component requirements, as well as to verify the implementation of software requirements and their components in the software architecture;
- low-level testing to verify compliance with low-level requirements.

After the production of the product, at the stage of conducting preliminary or interdepartmental tests, an assessment of the product's compliance with the requirements of the technical task regarding electromagnetic compatibility in working with on-board equipment is carried out [21-25].

When conducting experiments to assess the levels of electromagnetic interference created, the following are subject to investigation:

- voltage measurement of electromagnetic interference in the power supply wires of the product ;
- measurement of electromagnetic interference current strength in power supply circuits and in signal circuits;
- measurement of the electric field strength of the electromagnetic interference from different sides of the product.

When conducting experiments to assess the product's susceptibility to electromagnetic interference (EMF), the following are subject to investigation:

- resistance of the product to the influence of the magnetic field caused by the flow of alternating current in the inductor wire;
- resistance of the product to the influence of the magnetic field of sound frequency;
- resistance of the product to the impact of the electric field on the connecting cables of the product;
- resistance of the product to the influence of the fields of transient processes on the connecting cables of the product;

- resistance of the product to the influence of radio frequency electromagnetic interference on the power supply wires of the product;
- resistance of the product to the influence of radio frequency electromagnetic interference on the connecting cables of the product;
- resistance of the product to the effect of radio frequency radiation on the product and connecting cables;
- resistance of the product to the influence of electromagnetic interference of sound frequencies on the power supply wires of the product.

To ensure the product's resistance to external electromagnetic interference, the product uses specialized electroradio components that are resistant to elevated levels of EMF; materials and coatings that absorb EMF; structural elements of the grounding of the product and elements of metallization of its parts.

In particular, the following measures are taken to reduce the level of EMF emitted from harnesses and cables:

- shielding of circuits with impulse currents and the most important analog circuits with a low dynamic range;
- mutual compensation of magnetic fluxes created by these circuits due to the use of bifilar assembly;
- implementation of high-quality grounding of the shielding layer of harnesses and reduction of inductance during metallization of the shielding layer of harnesses and cables;
- reduction of the total resistance of the grounding bus;
- reduction of the area of the EMF radiation circuit by reducing the length of harnesses and cables;
- placement of harnesses and cables as close as possible to grounded structural elements: chassis, unit body, etc.;
- application of materials with frequency-dependent properties based on high-frequency ferrites for shielding of power circuits, circuits with impulse currents and the most important analog circuits with a low dynamic range.

In the part of designing a multilayer printed circuit board, to reduce the level of emitted EMF, the following rules must be followed:

- to increase the interlayer capacitance and ensure effective high-frequency separation, it is necessary that the power and ground layers are adjacent. Place the power supply polygons in the inner layers of the multilayer printed circuit board. Power supply layers should be made as continuous as possible, reducing the area of unmetallized areas;
- place all high-frequency circuits in the inner layers adjacent to the GND layers. Change the direction of the conductor track of high-speed signals in the topology of a multilayer printed circuit board in the form of an arc;
- the entire space of the printed circuit board, on which the components of the circuit and the communication line are not located, must be filled with landfill earth;
- to separate power supply buses of digital and analog circuits;
- input-output conductors of the printed circuit board must be made as short as possible and filtering of output signals should be provided.

To reduce the interference emission of the product, various circuit solutions are used, designed to suppress the spectral components of the main frequencies at which the product functions and their harmonics; specialized materials and coatings are used, which reduce the level of radiated EMF; structural elements of the grounding of the product and elements of metallization of its parts are used.

A series of experiments was conducted to evaluate the effectiveness of schematic and structural-technological solutions, which are the basis of the design of the BREO-class avionics product. The level of EMF emitted by the product in the frequency range of 0.01 MHz to 100 MHz was subject to evaluation.

The results of the experiments are shown in fig. 2.

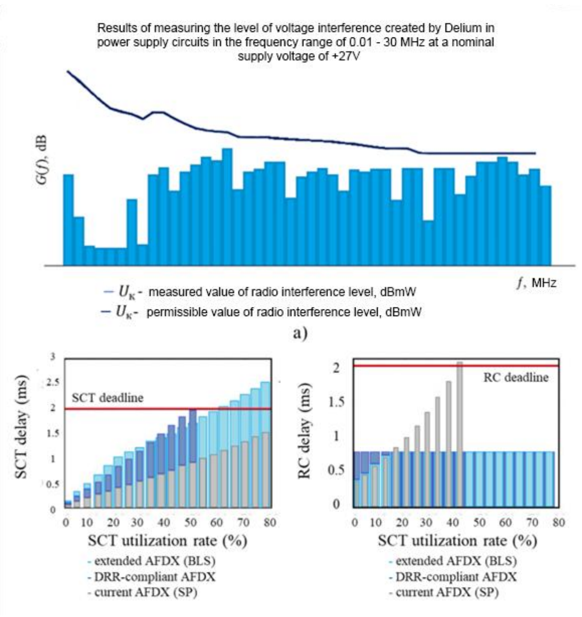

Fig. 2. The results of the experiments

Separately in fig. 2 a) shows the graph of the EPM level registered during the operation of the product in the frequency range of 0.01 MHz-30 MHz, in fig. 2 b) – in the frequency range of 30 MHz-100 MHz. Different lines on the graphs represent the level of registered EMFs and the permissible level that meets the requirements of regulatory and technical documentation for this class of equipment.

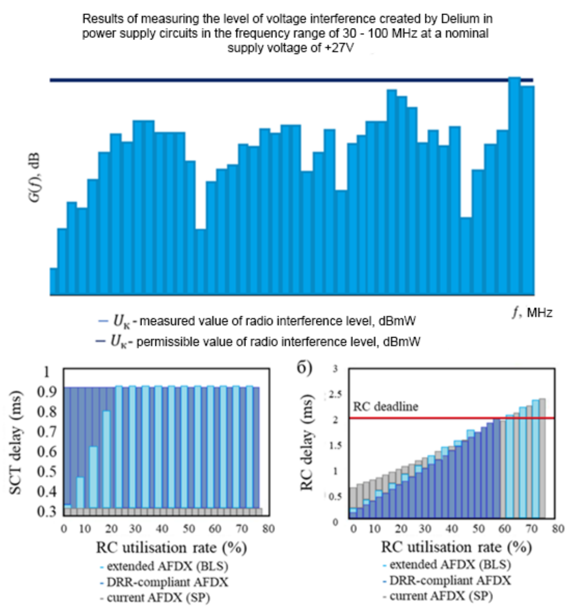

Fig. 3. Results of measuring the interference emission level of the product G (f) at nominal supply voltage + 27V in the frequency range f: a) 0.01 MHz-30 MHz, b) 30 MHz-100 MHz.

The presence of pronounced EMF pulsations in the low-frequency region is explained by the features of the product and is related to the clock frequency of the information exchange of the product with other users of bots of the complex through communication channels [26].

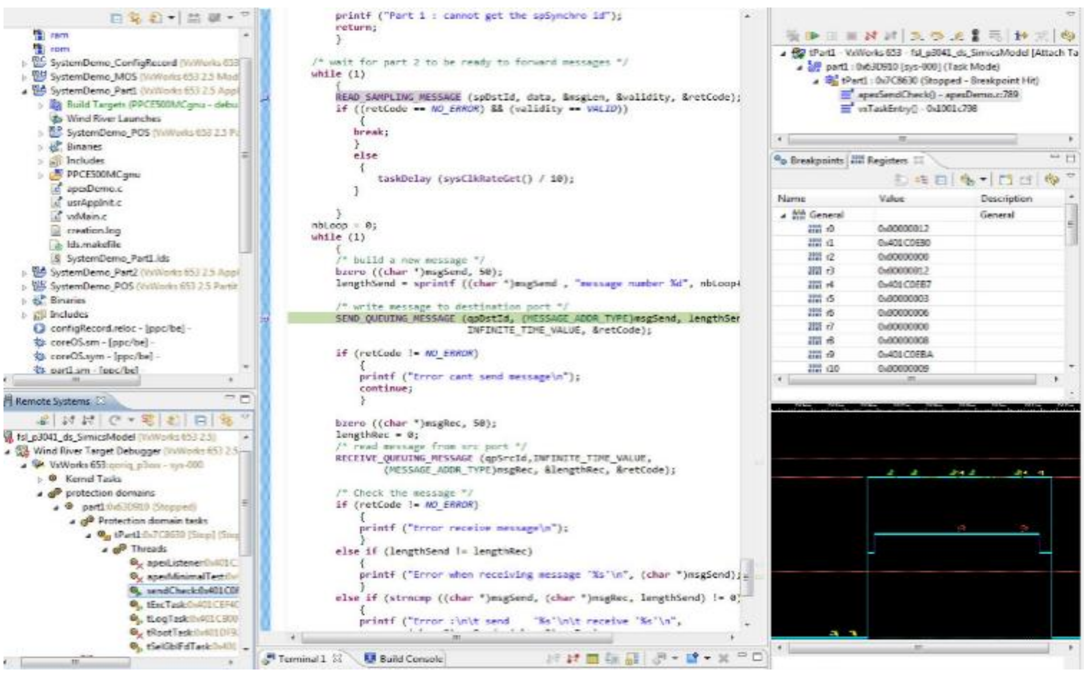

Fig. 4 Wind River Workbench showing VxWorks 653 partition debugging and Wind River System Viewer displaying ARINC partition behavior.

In the high-frequency region, the level of permissible EMFs is constant, and the pulsations of EMFs emitted by the product are due to the operation of the product at the clock frequency of the processor used in the product, with additional EMF generation at harmonic frequencies of the fundamental frequency. It is important to note that harmonics occur both at the main frequency of the product and at combined frequencies generated by the operation of the product nodes at frequencies other than the main one.

Conclusions

The proposed method and algorithm for controlling the computing system of integrated modular avionics during the flight of the aircraft for the control of functional elements and further development of the software of integrated modular avionics is important in the direction of ensuring the safety and reliability of aviation systems. This algorithm makes it possible to systematize and automate the process of controlling the functional elements of the computer system during the flight, which contributes to increasing the safety and efficiency of the flight. Algorithmic and software information support for the design and operation of onboard equipment of integrated modular avionics was developed for the verification of the created methods and models. They allow conducting experiments using developed methods and models, which contributes to the improvement of the quality and reliability of on-board equipment.

References

1. ARINC Specification 653. Avionics Application Software Standard Interface. URL: https://www.sae.org/standards/content/arinc653p3a-1/.

2. ARINC 664 Aircraft Data Networks. Part7: Avionics Full Duplex Switched Ethernet (AFDX) Network. Annapolis : Aeronautical Radio, Inc., 2005. 18 p.

1. ISO/IEC 7498-1:1994(E). Information technology. Open System Interconnection. Basic Reference Model: The Basic odel. Geneva : ISO/IEC, 1994. 62 p.

3. DO-178B Software Considerations in Airborne Systems and Equipment Certification. Dept. of Measurement and Information Systems. URL: https://inf.mit.bme.hu/sites/default/files/materials/taxonomy/term/445/13/13_CES_DO-178B.pdf.

4. RTCA DO-255 / EUROCAE ED-96. Requirements Specification for Avionics Computer Resource (ACR). URL: https: // standards. globalspec. com / std / 1968378 / RTCA% 20DO-255.

5. Burns A. Integrated Modular Avionics: Development, Deployment, and Integration. Chichester : Wiley, 2016. 320 p.

6. Dori D. Model-Based Systems Engineering with OPM and SysML. New York : Springer, 2016. 411 p.

7. Moir I., Seabridge A. Aircraft Systems: Mechanical, Electrical, and Avionics Subsystems Integration. 4th ed. Chichester : Wiley, 2013. 616 p.

8. Avizienis A., Randell B., Landwehr C. (eds.) Dependable Computing for Critical Applications. New York : Springer, 1991. 523 p.

9. Jones J. V. Reliability, Maintainability, and Supportability: Best Practices for Systems Engineers. 3rd ed. Hoboken : Wiley, 2006. 528 p.

10. Jukes M. Aircraft Data Monitoring Systems. Farnham : Ashgate, 2015. 384 p.

11. Rierson L. Safety Assessment of Software for Aircraft Systems. New York : CRC Press, 2013. 610 p.

12. Ducard G. J. Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles. London : Springer, 2009. 268 p.

13. Kopetz H. Real-Time Systems: Design Principles for Distributed Embedded Applications. 2nd ed. New York : Springer, 2011. 378 p.

14. Mukherjee A. S., Malshe A. K. Avionics Systems: Principles and Design. New Delhi : Hindustan Book Agency, 2009. 384 p.

15. Ghannem A. et al. Search-based requirements traceability recovery: A multi-objective approach. *2017 IEEE Congress on Evolutionary Computation (CEC)* : proceedings, Donostia, Spain, 05–08 June

2017 / IEEE. 2017. P. 1183–1190. DOI: https://doi.org/10.1109/CEC.2017.7969440.

16. Kim D., Lee S. H., Yang D. C. Fault diagnosis and monitoring of avionics systems based on integrated modular avionics architecture. *IEEE Access*. 2018. Vol. 6. P. 50185–50192. DOI: 10.1109/ACCESS.2018.2870811.

17. Cheng K., Tan Y., Zhao L. Reliability improvement in IMA systems through multi-level diagnostics. *Journal of Aerospace Information Systems*. 2019. Vol. 16(10). P. 456–465. DOI: 10.2514/1.I010737.

18. Liu Y., Jiang Z., Li X. Automated monitoring and diagnostic techniques for integrated modular avionics systems. *IEEE Transactions on Aerospace and Electronic Systems*. 2020. Vol. 56(4). P. 2938–2951. DOI: 10.1109/TAES.2020.297561.

19. Kim D., Lee S. H., Yang D. C. Fault diagnosis and monitoring of avionics systems based on integrated modular avionics architecture. *IEEE Access*. 2018. Vol. 6. P. 50185–50192. DOI: 10.1109/ACCESS. 2018.2870811.

20. Cheng K., Tan Y., Zhao L. Reliability improvement in IMA systems through multi-level diagnostics. *Journal of Aerospace Information Systems*. 2019. Vol. 16, no. 10. P. 456–465. DOI: 10.2514/1.I010737.

21. Liu Y., Jiang Z., Li X. Automated monitoring and diagnostic techniques for integrated modular avionics systems. *IEEE Transactions on Aerospace and Electronic Systems*. 2020. Vol. 56, no. 4. P. 2938–2951. DOI: 10.1109/TAES.2020.2975614.

22. Gao F., Wang H., Zhu C. A reliability assessment method for real-time avionics systems with diagnostics capability. *Chinese Journal of Aeronautics*. 2021. Vol. 34, no. 9. P. 1154–1164. DOI: 10.1016/j.cja.2020.11.004.

23. Kovalenko Y., Konakhovych H., Kozlyuk I. Species optimae perficiendi indicibus technicis operationis et adaequationis systematum electronicarum radiophonicae aircraft. *Acta inquisitionis et applicationis internationalis (IJERA)*. 2020. Vol. 10(09). P. 48–58. URL: www.ijera.com:10.9790/ 9622-1009044858

24. Kovalenko Y. Methods of developing integrated modular avionics systems. *Scientific Journal of Polonia University*. 2021. Vol. 43(6). P. 324–334. DOI: 10.23856/4341.

25. Kovalenko Y., Kozlyuk I. Reliability of computer structures of integrated modular avionics for hardware configurations. *System research and information technologies*. 2021. No. 2. P. 84–93. DOI: 10.20535/SRIT.2308-8893.2021.2.07.

26. Kovalenko Y. A Programmable Logic Controller (PLC); Programming Language Structural Analysis. *Advances in Intelligent Systems and Computing*. 2017. Vol. 543, P. 234–242, 2017. DOI: 10.1007/978-3-319-48923-0_29.

Kovalenko Y.B., Kozlyuk I.O.

METHOD OF DEVELOPING AUTOMATED INFORMATION SUPPORT FOR INTEGRATED MODULAR AVIONICS SYSTEMS: MONITORING, DIAGNOSTICS AND RELIABILITY IMPROVEMENT

Advances in avionics technology have significantly increased the complexity of modern aircraft systems. Integrated modular avionics (IMA) plays a critical role in providing an efficient, reliable, and maintainable avionics architecture by consolidating multiple functions onto common computing resources. This paper presents a method for developing automated information support software for IMA, focusing on real-time monitoring, diagnostics, and state management of avionics modules. The proposed approach emphasizes automation of data collection and processing, which enables early fault detection, predictive maintenance, and rapid response to component failures or degradation. By integrating automated monitoring capabilities, aircraft operators can improve system reliability, minimize downtime, and optimize maintenance planning. This paper highlights key cybersecurity considerations, including protection from external threats, intrusion detection, and secure data transfer between IMA *modules. As networked avionics systems become increasingly prevalent, implementing robust security measures is essential to prevent unauthorized access, data manipulation, or system failures that could jeopardize flight safety. Additionally, this study examines key aspects of system architecture design, emphasizing modularity, scalability, and compliance with industry standards such as ARINC 653 and DO-178C. The modular nature of IMA provides greater flexibility in system design, allowing new features to be integrated or existing modules to be upgraded without extensive reconfiguration. Compliance with aviation safety standards and software certification ensures that automated information support solutions meet regulatory requirements. Recommendations for improving diagnostic and monitoring algorithms are provided to optimize system performance. Advanced data analytics techniques, including machine learning and predictive analytics, are explored to improve fault detection accuracy and optimize maintenance schedules. Using real-time data analytics, operators can implement condition-based maintenance strategies, reducing unnecessary inspections while improving overall system reliability. The results of this study confirm that the integration of automated information support systems significantly improves the efficiency and reliability of IMA-based avionics. By implementing advanced monitoring and diagnostic capabilities, these systems contribute to increased aircraft availability, reduced maintenance costs, and improved flight safety. In addition, by ensuring compliance with modern aviation standards, such solutions align with industry best practices and regulatory requirements. In conclusion, the development and implementation of automated information support software for IMA represents an important step towards achieving higher levels of operational efficiency, reliability, and cybersecurity in modern aviation. As avionics systems continue to evolve, the integration of intelligent monitoring solutions will be essential to meet the growing demands of next-generation aircraft while ensuring compliance with stringent safety and certification standards.*

***Keywords**: integrated modular avionics (IMA); automated information support; real-time monitoring; diagnostics; system architecture; service efficiency.*

Коваленко Ю.Б., Козлюк І.О.

МЕТОД РОЗРОБКИ АВТОМАТИЗОВАНОГО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕГРОВАНИХ МОДУЛЬНИХ СИСТЕМ АВІОНІКИ: МОНІТОРИНГ, ДІАГНОСТИКА ТА ПІДВИЩЕННЯ НАДІЙНОСТІ

Досягнення в авіоніці значно модернізували сучасні системи повітряних суден. Інтегрована модульна авіоніка (IMA) відіграє вирішальну роль у забезпеченні ефективної, надійної та зручної архітектури авіоніки шляхом консолідації кількох функцій на загальних обчислювальних ресурсах. У статті представлено метод розробки автоматизованого програмного забезпечення інформаційної підтримки для IMA, зосереджуючись на моніторингу в реальному часі, діагностиці та управлінні станом модулів авіоніки. Запропонований підхід наголошує на автоматизації збору та обробки даних, що дозволяє раннє виявлення несправностей, прогнозоване технічне обслуговування та швидке реагування на збої компонентів або погіршення якості. Інтегруючи можливості автоматизованого моніторингу, оператори повітряних суден можуть підвищити надійність системи, мінімізувати час простою та оптимізувати планування технічного обслуговування. У статті висвітлюються ключові аспекти кібербезпеки, зокрема захист від зовнішніх загроз, виявлення вторгнень і безпечна передача даних між модулями IMA. Оскільки мережеві авіонічні системи стають все більш поширеними, впровадження надійних заходів безпеки є важливим для запобігання несанкціонованому доступу, маніпулюванню даними або системним збоям, які можуть загрожувати безпеці польотів. Крім того, це дослідження вивчає ключові аспекти проектування архітектури системи, наголошуючи на модульності, масштабованості та відповідності галузевим стандартам,

таким як ARINC 653 і DO-178C. Модульний характер IMA забезпечує більшу гнучкість у проектуванні системи, дозволяючи інтегрувати нові функції або оновлювати існуючі модулі без значної реконфігурації. Відповідність стандартам авіаційної безпеки та сертифікація програмного забезпечення гарантує відповідність рішень автоматизованої інформаційної підтримки нормативним вимогам. Для оптимізації продуктивності системи надаються рекомендації щодо вдосконалення алгоритмів діагностики та моніторингу. Передові методи аналізу даних, включаючи машинне навчання та прогнозну аналітику, досліджуються для підвищення точності виявлення несправностей і оптимізації графіків технічного обслуговування. Використовуючи аналітику даних у режимі реального часу, оператори можуть впроваджувати стратегії технічного обслуговування на основі стану, зменшуючи непотрібні перевірки та підвищуючи загальну надійність системи. Результати цього дослідження підтверджують, що інтеграція автоматизованих систем інформаційної підтримки значно підвищує ефективність і надійність авіоніки на основі IMA. Впроваджуючи розширені можливості моніторингу та діагностики, ці системи сприяють підвищенню доступності літака, зниженню витрат на технічне обслуговування та підвищенню безпеки польотів. Крім того, забезпечуючи відповідність сучасним авіаційним стандартам, такі рішення відповідають найкращим галузевим практикам і нормативним вимогам. Отже, розробка та впровадження програмного забезпечення автоматизованої інформаційної підтримки для IMA є важливим кроком на шляху до досягнення вищих рівнів оперативної ефективності, надійності та кібербезпеки в сучасній авіації. Оскільки системи авіоніки продовжують розвиватися, інтеграція інтелектуальних рішень для моніторингу буде важливою для задоволення зростаючих вимог до літаків наступного покоління, забезпечуючи при цьому відповідність суворим стандартам безпеки та сертифікації.

Ключові слова: *інтегрована модульна авіоніка (IMA); автоматизована інформаційна підтримка; моніторинг в реальному часі; діагностика; архітектура системи; ефективність обслуговування.*